

Postel

Network Working Group
Request for Comments # 54
June 18, 1970

Steve Crocker (UCLA)
Jon Postel (UCLA)
John Newkirk (Harvard)
Mike Kraley (Harvard)

An Official Protocol Proffering

I. INTRODUCTION

As advertised in NEW/RFC #53, we are submitting the protocol herein for criticism, comments, etc. We intend for this protocol to become the initial official protocol, and will, therefore, be happiest if no serious objections are raised. Nevertheless, we will entertain all manner of criticism until July 13, 1970, and such criticism should be published as a NWG/RFC or directed to the first author.

After July 13, a decision will be made whether to adopt this protocol (or a slight variation) or whether to redesign it and resubmit it for criticism.

Only the Protocol

In preceding discussions of protocol, no clear distinction has been made between the network-wide specifications and local strategies. We state here that the only network-wide issues are message formats and restrictions on message content. Implementation of a Network Control Program (NCP) and choice of system calls are strictly local issues.

This document is constrained to cover only network-wide issues and thus will not treat system calls or NCP tables; nevertheless, a protocol is useless without an NCP and a set of system calls, so we have expended a great deal of effort in deriving a prototypical NCP. This effort is reported in NWG/RFC #55, and the reader should correlate the protocol presented here with the suggestions for using it presented there. It is important to remember, however, that the content of NWG/RFC #55 is only suggestive and that competitive proposals should be examined before choosing an implementation.

Flow Control

In the course of designing this current protocol, we have come to understand that flow control is more complex than we imagined. We now believe that flow control techniques will be one of the active areas of concern as the network traffic increases. We have, therefore, benefitted from some ideas stimulated by Richard Kaline and Anatol Holt and have modified the flow control procedure. (Defects in our scheme are, of course, only our fault.) This new procedure has demonstrable limitations, but has the advantages that it is more cleanly implementable and will support initial network use. This is the only substantive change from the protocol already agreed upon.

The new flow control mechanism requires the receiving host to allocate buffer space for each connection and to notify the sending host of how much space in bits is available. The sending host keeps track of how much room is available and never sends more text than it believes the receiving host can accept.

To implement this mechanism, the sending host keeps a counter associated with each connection. The counter is initialized to zero, increased by control commands sent from the receiving host, and decremented by the text length of any message sent over the connection. The sending host is prohibited from sending text longer than the value of the

counter, so the counter never goes below zero.

Ideally, the receiving host will allocate some buffer space as soon as the connection is established. The amount allocated must never exceed what the receiver can guarantee to accept. As text arrives, it occupies the allocated buffer space. When the receiving process absorbs the waiting text from the buffer, the NCP fires back a new allocation of space for that connection. The NCP may allocate space even if the receiving process has not absorbed waiting text if it believes that extra buffer space is appropriate. Similarly, the NCP may decide not to reallocate buffer space after the receiving process makes it available.

The control command which allocates space is

ALL <link> <space>

This command is sent only from the receiving host to the sending host.

This formulation of flow control obviates the RSM and SPD commands in MWG/RFC #36, and the Host-to-Imp message type 10 and Imp-to-Host message types 10 and 11 in the current revision of BBN Report 1822.

The obvious limitation in this scheme is that the receiving host is not permitted to depend upon average buffer usage--worst case is always assumed. If only a few connections are open, it is unlikely that there would be much savings. However, for more than a few connections, average buffer usage will be much less than allocated buffer space. We have looked at extensions of this protocol which would include adaptive allocation, and we believe this to be feasible. For the present this limited scheme seems best, and we look forward to discussing more sophisticated schemes later. The old scheme of special RFSM's, etc. also remains under discussion.

In order to answer questions and discuss details, we will hold a pair of network meetings. The first will be on June 29 at Harvard and the second on July 1 at UCLA. We request that no more than one programmer per host attend a meeting and that hosts be represented at only one of these meetings. Two of us (J.N. and S.C.) will be at both meetings.

To make reservations to attend the Harvard meeting, contact

Mrs. Margi Robison

(617) 495-3989

or 495-3991

To make reservations to attend the UCLA meeting, contact Mrs.

Benita Kirstel (213) 825-2368.

II. THE PROTOCOL

The notion of a connection as explained in NWG/RFC #33 pervades the protocol. A connection is a simplex communication path, intended to be between two processes.

The primary function of the protocol is to provide for

- (1) establishment of connections,
- (2) regulation of flow over connections, and
- (3) termination of connections.

In addition, the protocol provides some ancillary functions such as sending simulated interrupt pulses and echoing test messages.

To provide a path for exchanging information about connections, we designate specific links, i.e. link one between each pair of hosts to be control links. Traffic on control links consists only of control commands, defined below.

Connections are named by a pair of sockets. Sockets are 40 bit names which are known throughout the network. Each host is assigned a private subset of these names, and a command which requests a connection names one socket which is local to the requesting host and one local to the receiver of the request.

Sockets are polarized; even numbered sockets are receive sockets; odd numbered ones are send sockets. One of each is required to make a connection.

To facilitate transmission of information over a connection, a unique link is assigned to each connection. One of the steps in establishing a connection, therefore, is the assignment of a link. Of the non-control links, zero is reserved for intra-network use, and links 32 to 255 are reserved for experiment and expansion. Thus only links 2 through

The nature of a connection is explained in terms of the
the process. A connection is a single continuous line, intended
to be between two processes.

The primary function of the connection is to provide the

(1) establishment of connection.

(2) regulation of flow over connection, and

(3) termination of connection.

In addition, the connection provides some auxiliary functions such

as holding material in place and acting as a support.

To provide a basis for separating different or distinct functions,

we distinguish specific links, i.e., link one between two parts of a body.

As the control link, the link is defined as any of several

members, defined below.

Connections are made by a pair of members. Members are of two

kinds which are shown in the figure. Each part is assumed to

provide support of other parts, and a support which requires a connection

means one member which is fixed to the supporting part and one fixed to

the member of the support.

Members are indicated by arrows and are shown in the figure.

and numbered from one to ten. One of two is required to make

a connection.

To facilitate the understanding of information over a connection, a

control link is assigned to each connection. One of the links is assigned

ing a connection, therefore, is the assignment of a link. Of the two

control links, one is assigned to the first-order link and link 11 is

111 are reserved for expansion and explanation. Thus, only link 11 is

31 are available for regular use. Link assignment must either always be done by the receiver or always by the sender. We have (almost) arbitrarily chosen this to be the receiver's responsibility.

All regular messages consist of a 32 bit leader, marking, text, and padding. Marking is a (possibly null) sequence of zeroes followed by a 1; padding is a 1 followed by a (possibly null) sequence of zeroes.

A regular message sent over the control link (link 1) is called a control message. Its text is an integral (possibly zero) number of control commands in the form described below, and this text must end on a command boundary.

The commands used to establish a connection are STR and RTS. The STR command is sent from a prospective sender to a prospective receiver. Its <my socket> field contains a send socket local to the prospective sender; its <your socket> field contains a receive socket local to the prospective receiver. The RTS command is the dual, but it also contains a <link> field for link assignment. These two commands are referred to as requests-for-connection (RFC). A STR and an RTS match if the <my socket> field of one is identical to the <your socket> field of the other and vice versa. A connection is established where a matching pair of RFC's have been exchanged.

Hosts are prohibited from establishing more than one connection to any local socket. Therefore, a host may not use a socket for the <my socket> field of an RFC if that socket is mentioned in a previous RFC and the connection is not yet terminated.

The command used to terminate a connection is CLS. Each side must send and receive a CLS command before a connection is completely terminated and the sockets are free to participate in other connections. It

It is available for regular use. This alignment must also be

be done by the user or by the system. The user (client)

must be given a responsibility.

All regular messages consist of a 32 bit header, body, and

and padding. The header is a 32-bit field which contains the

by a 32-bit field which contains the possibly null sequence of

A regular message sent over the network also has a 32-bit

a regular message. The first 32-bit field contains the

control commands in the form described below, and the last 32-bit

a regular message.

The control part of a regular message is a 32-bit field. The

the control is sent from a perspective sender to a perspective

the perspective field contains a 32-bit field in the perspective

sender the perspective field contains a 32-bit field in the

perspective sender. The 32-bit field is the field, but it also

a 32-bit field for the perspective. These two fields are

as perspective-for-perspective (PFP). A 32-bit field is the

perspective field of the perspective field of the perspective

and vice versa. A connection is established when a regular

the perspective field is established.

These are perspective field establishing with the perspective

any local sender. Therefore, a 32-bit field is the

perspective field of the perspective field of the perspective

and the connection is not yet established.

The control part of a regular message is a 32-bit field. The

the control is a 32-bit field which contains a 32-bit

field and the perspective field of the perspective field of the

is not necessary that both RFC's be exchanged before a connection is terminated. More details on termination are given below.

After a connection is established, the receiving host sends a ALL command which allocates space for the connection. The sender keeps track of how much space is available in the receiving host and does not transmit more text than the receiving host can accept, as explained above. A sender is also constrained by the local IMP from sending a message over a connection until the RFNM from the previous message is received.

After a connection is established, CLS commands sent by the receiver and sender have slightly different effects. CLS command sent by the sender indicate that no more messages will be sent over the connection. This command must not be sent if there is a message in transit over the connection.

CLS commands sent by the receiver act as demands on the sender to terminate transmission. However, since there is a delay in getting the CLS command to the sender, the receiver must expect its buffers to fill to the limit provided in ALL commands.

While a connection is established, either side may send INR or INS commands. The interpretation of these commands is a local matter, but in general they will provide an escape function.

Note that the ALL, INR and INS commands may be sent only after the connection is established and before a CLS command is sent.

A very simple test facility is provided by the ECO and ERP commands. Upon receiving a ECO command, a host must change the first eight bits to ERP and return it. These commands have no relationship to connections.

A NOP command is included for convenience. It is coded as zero to facilitate command message construction.

is not necessary that both are in the same position.

Furthermore, the details of the operation are given below.

After a connection is established, the receiving party sends a call

command which indicates the space for the connection. The sender sends

of how much space is available in the receiving party and how much

the sender sends when the receiving party can accept an additional space.

A sender is also notified by the local SW that sending a message

over a connection will be done from the provided number in the

After a connection is established, the sender sends by the receiver

and sender have slightly different effects. The sender sends by the

sender indicates that no more messages will be sent over the connection.

This command will not be sent if there is a message in transit over the

connection.

The sender sends by the receiver and the sender to the receiver

to indicate the connection. However, since there is a delay in getting

the SW command to the sender, the receiver will expect the sender to

fill in the last provided in the command.

While a connection is established, either side may send the SW

command. The interpretation of these commands is a little subtle, but

in general they will generate an empty function.

Note that the SW, SW and SW commands may be sent only after the

connection is established and before a SW command is sent.

A very simple test facility is provided by the SW and SW commands.

When receiving a SW command, it will send the SW test light SW to

SW and return SW. These commands have no relationship to connections.

A SW command is included in the SW command. It is coded as SW to

facilitate command message construction.

Finally, an ERR command is included for notifying a foreign host it has (apparently) made an error. At present, no specific list of errors is defined, and no action is defined for the receipt of ERR commands. Hosts should log ERR commands upon receipt so that system programmers can diagnose the trouble. A host may generate an ERR command at any time and for any reason, but it is advised that each host publish an exhaustive list of the ERR commands it may send and their interpretations.

NETWORK CONTROL COMMANDS

The following is a detailed description of the structure and format of each of the control commands.

To facilitate and clarify socket descriptions, the following conventions have been adopted:

<my socket> and <your socket> are used in command descriptions.

<my socket> is local to the originator of the command.

<your socket> is local to the receiver of the command.

Finally, as the document is intended for public use, it is necessary to ensure that the information is presented in a clear and concise manner.

It is important to note that the information presented in this document is for informational purposes only and should not be used for any other purpose.

Where it is necessary to include information that is not directly related to the main topic, it should be included in a separate section.

It is also important to ensure that the information is presented in a way that is easy to understand and that it is presented in a logical order.

Finally, it is important to ensure that the information is presented in a way that is consistent with the overall purpose of the document.

For any questions or comments, please contact the author at the address listed below.

List of the information to be included in the document:

THE INFORMATION TO BE INCLUDED

The following is a list of the information to be included in the document:

1. The title of the document.

2. The author's name and address.

3. The date of the document.

4. The purpose of the document.

5. The scope of the document.

6. The conclusions of the document.

CONTROL COMMAND FORMATS

No Operation

NOP

Request Connection, Receiver to Sender

RTS	my socket	your socket	link
-----	-----------	-------------	------

Request Connection, Sender to Receiver

STR	my socket	your socket
-----	-----------	-------------

Close

CLS	my socket	your socket
-----	-----------	-------------

Allocate

ALL	link	space
-----	------	-------

. Interrupt, Sent by Receiving Process

INR	link
-----	------

Interrupt Sent by Sending Process

INS	link
-----	------

no results

100

Request Connection, Receiver to Sender

100	my contact	your contact	100
-----	------------	--------------	-----

Request Connection, Sender to Receiver

100	my contact	your contact
-----	------------	--------------

Close

100	my contact	your contact
-----	------------	--------------

Allow

100	link	space
-----	------	-------

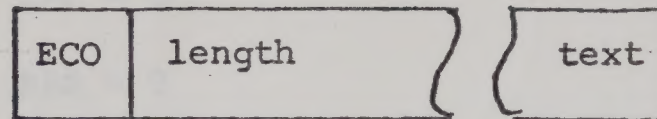
Integrity test of existing process

100	link
-----	------

Integrity test of existing process

100	link
-----	------

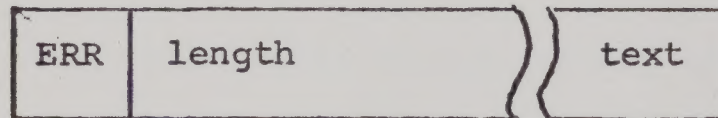
Echo Request



Echo Reply



Error Detected



The host is specified in the leader.

<link> is 8 bits

<space> is 32 bits long and is an unsigned integer.

<length> is an unsigned 16 bit integer.

<text> is as long as the length. The command is therefore 24 bits longer than the length. Maximum length is one message, to facilitate command decoding and manipulation.

All control command codes are 8 bits long:

NOP = 0

RTS = 1

STR = 2

CLS = 3

ALL = 4

INR = 5

INS = 6

ECO = 7

ERP = 8

ERR = 9

<my socket> and <your socket> are 32 bits long,

User number	AEN
-------------	-----

24 bits for user number and 8 bits for AEN.

1000 = 1

1000 = 2

1000 = 3

My address and phone number are as follows:

First name	Mr.
------------	-----

Is this for your number and 5 lines for Mr.

III. Conclusion

Extensions to the Protocol

Some issues have not been adequately treated in the current protocol. We have in mind the following topics to consider more thoroughly and perhaps experiment with.

1. More Sophisticated Flow Control.

As mentioned above, other schemes for flow control are still being considered. Other than the necessity of providing some form of it, we are completely unsure of the nature of the problem. It may turn out that the present scheme is completely adequate; it may also turn out that we will need a much more complex scheme.

2. Error Detection and Recovery

As we gain some experience with the network, we will develop a better understanding of what errors can occur and, perhaps more importantly, what to do about these errors. We expect the protocol to change as we understand error control.

3. Start Up and Shut Down Procedures

We have not done enough thinking about the problem of the host which participates part-time in the network, which ceases normal network operation but remains on the network for special purposes, or

Expenditures in the Program

From 1955 to 1956 the total expenditures in the program

amounted to \$1,000,000. The following table shows the distribution of

expenditures by category and percentage of total.

I. Basic Research and Development

As mentioned above, other activities for 1955

amounted to \$1,000,000. Other than

the necessary operating costs of \$1,000,000

are completely absorbed by the cost of the

program. It was found that the present system

is completely adequate. It was found that the

will need a new computer system.

II. Basic Research and Development

In the past some expenditures were for

networks, and will be for a better understanding

of what errors are caused and, perhaps, some in-

formation, which is to be used in the future.

Except the network, no change is to be made.

Other things.

III. Basic Research and Development

It was found that enough information about the

problem of the last which pertains to the

the network, which means some more information

and results of the network, the network, or

which recovers from a system failure. These issues are critical to robust network operation and are possibly our highest priority.

4. Query and Response

A host-to-host status test would be a valuable tool, but it is not yet clear what is appropriate to provide.

which involves first a system failure. These failures
are critical to current network operation and are
possibly our highest priority.

4. Query and response

A host-to-host session would be a vital
link, but it is not yet clear what is required.
Please see appendix.

Coming onto the Network

We suggest that hosts come onto the network gingerly. First, each host should thoroughly exercise connections to itself. Then it should arrange experiments with some other host who is already functioning. Finally, it may begin to exercise the facilities of other hosts. It is not clear at this time which host will be in the best position to help other hosts first, but UCLA will attempt to serve this function.

Private Networking

A common ploy is to use the IMP to connect several local computers, one or more of which is not available to the whole network. For example, Harvard is connecting its PDP-1 to its PDP-10 via an IMP; Lincoln Laboratories is connecting its TSP to the 360/67 and the TX2 via an IMP; and UCLA is similarly connecting a XDS 920 to its Sigma-7. In each of these cases, the small machine will not initially provide services to the network.

Although there should be no unwanted traffic to any of these extra hosts, it is desirable that they conform minimally to the network protocol. Provided that they never initiate a connection or send out spurious control commands, it is sufficient for a host to respond to CLS commands with acknowledging CLS commands, and to respond to ECO commands with ERP commands.

We suggest that these two are the network's primary functions. Each has a different emphasis: the first is to learn the network's internal structure, while the second is to learn the network's external structure. This is a natural division of labor. It is not clear at this time which part will be the most important in helping the network learn, but both will be important in solving this problem.

Network Learning

A common way to learn the network's internal structure is to use a learning algorithm. This is a process by which the network's internal structure is modified so that it can learn from the data. The network's internal structure is modified by changing the weights of the connections between the nodes. The weights are initially set to random values, and the network is trained by presenting it with a set of data. The network's internal structure is modified until it can learn from the data. This is a process of trial and error, and it can take a long time to learn the network's internal structure.

Although there are many different learning algorithms, they all have the same basic idea: the network's internal structure is modified until it can learn from the data. This is a process of trial and error, and it can take a long time to learn the network's internal structure. The network's internal structure is modified by changing the weights of the connections between the nodes. The weights are initially set to random values, and the network is trained by presenting it with a set of data. The network's internal structure is modified until it can learn from the data. This is a process of trial and error, and it can take a long time to learn the network's internal structure.

Acknowledgments

The work presented above is only a small portion of the concurrent effort. Most of the related effort will be reported in immediately forthcoming RFC's. A number of people provided extremely valuable aid during the last two weeks. We are particularly grateful to Professor George Mealy of Harvard for supporting four of his students to come westward to work on the network, to Robert Uzgalis for facilitating access to CCN at UCLA, and to the secretarial staff of the Computer Science Division of the University of Utah, and especially Peggy Tueller and Marcella Sanchez, for excellent help in preparing these documents.

